arXiv:1301.7693v2 [cs.IT] 1 Feb 2013

Optimal Locally Repairable Codes and Connections to Matroid Theory

Itzhak Tamo[*†], Dimitris S. Papailiopoulos[‡] and Alexandros G. Dimakis[‡]
[*] Dept. of ECE and Inst. for Systems Research University of Maryland, USA
[†]Electrical and Computer Engineering, Ben-Gurion University of the Negev, Israel
[‡] Electrical and Computer Engineering University of Texas at Austin, USA
tamo@umd.edu, dimitris@utexas.edu, dimakis@austin.utexas.edu

***Abstract*—Petabyte-scale distributed storage systems are currently transitioning to erasure codes to achieve higher storage efficiency. Classical codes like Reed-Solomon are highly suboptimal for distributed environments due to their high overhead in single-failure events. *Locally Repairable Codes* (LRCs) form a new family of codes that are repair efficient. In particular, LRCs minimize the number of nodes participating in single node repairs while generating small network traffic for repairs. Two large-scale distributed storage systems have already implemented different types of LRCs: Windows Azure Storage and the Hadoop Distributed File System RAID used by Facebook. The fundamental bounds for LRCs, namely the best possible distance for a given code locality, were recently discovered, but few explicit constructions exist. In this work, we present an explicit and simple to implement construction of optimal LRCs, for code parameters previously established only by existence results. For the analysis of the code's optimality, we derive a new result on the matroid represented by the code's generator matrix.**

I. Introduction

Traditional architectures of large-scale storage systems rely on distributed file systems that provide reliability through block replication. Typically, data is split in blocks and three copies of each block are stored in different storage nodes. The major disadvantage of triple replication is the large storage overhead. As the amount of stored data is growing faster than hardware infrastructure, this becomes a factor of three in the *storage growth rate*, resulting in a major data center cost bottleneck.

As is well-known, *erasure coding* techniques achieve higher data reliability with considerably smaller storage overhead [1]. For that reason different codes are being deployed in production storage clusters. Application scenarios where coding techniques are being currently deployed include cloud storage systems like Windows Azure [2], big data analytics clusters (*e.g.*, the Facebook Analytics Hadoop cluster [3]), archival storage systems, and peer-to-peer storage systems like Cleversafe and Wuala.

It is now understood that classical codes (such as Reed-Solomon) are highly suboptimal for distributed storage repairs [4]. For example, the Facebook analytics Hadoop cluster discussed in [3] deployed Reed-Solomon encoding for 8% of the stored data. That portion of the data generated repair traffic approximately equal to 20% of the total network traffic. Therefore, as discussed in [3], the main bottleneck in increasing code deployment in storage systems is designing new codes that perform well for distributed repairs.

Three major repair cost metrics have been identified in the recent literature: *i)* the number of bits communicated in the network, *i.e.*, the *repair-bandwidth* [4]–[9] *ii)* the number of bits read, the *disk-I/O* [7], [10] and *iii)* more recently the number of nodes that participate in the repair process, also known as, *repair locality*. Each of these metrics is more relevant for different systems and their fundamental limits are not completely understood. In this work, we focus on the metric of repair locality, one that seems most relevant for single-location high-connectivity storage clusters.

Locality was identified as a good metric independently by Gopalan *et al.* [11], Oggier *et al.* [12], and Papailiopoulos *et al.* [13]. Consider a code of total length n with k information symbols. Symbol i has locality r_i if it can be reconstructed by accessing at most r_i other code symbols. For example, in an (n,k) MDS code, every symbol has trivial locality k. We will say that a systematic code has *information symbol locality* r if all the k information symbols have locality r. Similarly, we will say that a code has *all-symbol locality* r if all n symbols have locality r. Codes that have good locality properties were initially studied in [14], [15].

In [11], a trade-off between code distance, *i.e.*, reliability, and information symbol locality was derived for scalar linear codes. In [16], an information theoretic trade-off for any code (linear/nonlinear) was derived when considering all symbol locality. An (n,k) code with (information symbol or all-symbol) locality r has minimum distance d that is bounded as

$$d \leqslant n-k-\left\lceil \frac{k}{r} \right\rceil + 2. \tag{1}$$

Bounds on the code-distance for a given locality were also derived and generalized in parallel and subsequent works [17]–[19].

An (n,k,r) *locally repairable code* (LRC) is an (n,k) code such that *any* of its symbols can be reconstructed by accessing and processing *at most* r other symbols (all-symbol locality). Codes with all-symbol locality that meet the above bound are termed *optimal LRCs* and are known to exist when $(r+1)$ divides n [11], [16]–[19]. Explicit optimal LRC constructions for some code parameters were introduced in [16]–[20]. Some works extend the designs and theoretic bounds to the case where repair bandwidth and locality are jointly optimized under multiple local failures [18], [19] and/or security issues are addressed [19]. The construction of practical LRCs is further motivated by the fact that two major distributed storage systems have already implemented different types of LRCs: Windows Azure Storage [2] and the Hadoop Distributed File System RAID used by Facebook [3]. As of now, designing LRCs with optimal distance for most code parameters n, k, r that are easy to implement was left as a new and exciting open problem.

Our Contribution: We introduce a new *explicit* family of optimal (n, k, r)-LRCs. Our construction is optimal for any (n, k, r) such that $r+1$ divides n. Our codes require $O(k \log n)$ bits in the description of each symbol and their main advantage is in their design. The codes are very simple to implement and are based on Reed-Solomon codes with added symbols that account for locality. The main theoretical challenge is in proving that they are optimal. This is done by first establishing a connection between the minimum distance and locality of a linear code and the matroid that it represents. This connection will provide a sufficient condition for optimal LRCs. Then it is shown using some properties of the determinant function and polynomials over finite fields, that the code generator matrix satisfies this condition.

The most related works to ours are the two parallel and independent studies of [18] and [19]. There, optimal LRC constructions for similar range of code parameters are presented. Although these constructions rely on different tools and designs than the ones presented here, it would be of interest to explore further connections.

The remainder of the paper is organized as follows. In Section II, we present our code construction. In Section III, we establish a precise formula of the minimum distance of a linear code in terms of the matroid represented by the generator matrix. In Section IV, we use the established results and algebra of polynomials over finite fields to prove the optimality of our code.

II. Code Construction

In this section we construct an *optimal* (n, k, r)-LRC for the following sets of coding parameters

$$r+1 \text{ divides } n \tag{2}$$

$$\text{or } n \bmod (r+1) - 1 \geqslant k \bmod r > 0. \tag{3}$$

due to space limitation we prove its optimality only for the case where $r+1$ divides n. Let $m = n\frac{r}{r+1}$ and assume that $1 < r < k$.[1]

Our construction is very simple: we take the output of an (m, k)-Reed-Solomon code and for each r coded symbols we re-encode them into $r+1$ symbols using a *specific* MDS code. This simple construction will be shown to have *i)* the desired locality r and *ii)* optimal minimum distance. In Fig.1, we give a sketch of our construction.

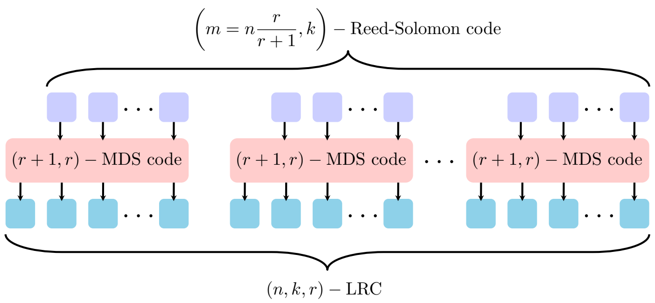

Figure 1. A sketch of our (n, k, r)-LRC construction. We start with an $(m = n\frac{r}{r+1}, k)$-Reed-Solomon code. We then re-encode the m Reed-Solomon coded blocks in the following manner: each consecutive group of r coded blocks is re-encoded using a specific $(r+1, r)$ MDS code. The n outputs of the $\frac{m}{r}$ local codes are the encoded blocks of our LRC. It should not be hard to see that the locality r can be trivially obtained by the local codes: if a block is missing the remaining r coded blocks in its group can be used to reconstruct it. Although our code as presented in this figure is not in systematic form, it can be easily done so by a simple transformation of the generator matrix.

More formally, Let $\mathbb{F}_p$ be a field of size $p \geqslant m$, and consider a file that is cut into k blocks $x = [x_1, \ldots, x_k]$, where each block is an element of the field $\mathbb{F}_{p^{k+1}}$. These k blocks are encoded into n coded-blocks $y = [y_1, \ldots, y_n]$:

$$y = x \cdot G$$

where G is the generator matrix defined over the field extension $\mathbb{F}_{p^{k+1}}$. The construction of G follows.

Construction 1 *Let $\alpha_1, \ldots, \alpha_m$ be m distinct elements of the field $\mathbb{F}_p$, with $p \geqslant m$, and a be a primitive element of the field $\mathbb{F}_{p^{k+1}}$. Also let V be a $k \times m$ Vandermonde matrix with its i-th column being equal to $\overline{\alpha}_i = (1, \alpha_i, \ldots, \alpha_i^{k-1})^t$. Then, the generator matrix of the code is*

$$G = V \cdot (I_{m/r} \otimes A),$$

where I_s is the identity matrix of size s and $A = (a_{i,j})$ is an $r \times (r+1)$ matrix defined as follows: ones on the main diagonal, and a's on the diagonal whose elements $a_{i,j}$ satisfy $j - i = 1$.

An example of an $(r+1 = 4, r = 3)$ A matrix is given bellow

$$A = \begin{pmatrix} 1 & a & 0 & 0 \\ 0 & 1 & a & 0 \\ 0 & 0 & 1 & a \end{pmatrix}.$$

Notice that the matrix A serves as the generator matrix of the $(r+1, r)$ MDS code used in the second encoding step. This step provides the locality property of the code.

Notice that G is not in systematic form: no k subsets of its columns form the identity matrix. However, there is an easy way to do so, by retaining the locality and distance properties: pick k linearly independent columns of G, say G_k, and use as a new code generator matrix the matrix $G_{\text{sys}} = G_k^{-1} G$.

Theorem 1 *The code generated by G has locality r and optimal minimum distance $d = n - k - \left\lceil \frac{k}{r} \right\rceil + 2$, when $(r+1)|n$.*

[1] If $r = k$ any (n, k) MDS code is an optimal $(n, k, r = k)$ LRC. Moreover, if $r = 1$ we can show that since r divides k then $r + 1 = 2$ has to divide n, i.e. n is even. The duplication of each symbol twice in an $(n/2, k)$ MDS code will result in an optimal $(n, k, r = 1)$ LRC.

The proof of the above theorem is done in two steps. First, in Section III, we derive a new result that expresses the minimum distance of a linear code using the matroid represented by its generator matrix. This result will imply that it is sufficient to verify that some subsets of k columns of G are full-rank. Then, in Section IV, we show that these submatrices are indeed invertible, by using properties of the determinant function and polynomials over finite fields.

III. Matroids and Locally Repairable Codes

A. Overview of Matroid Theory

We start with a quick overview of Matroid Theory. A matroid $\mathcal{M} = \mathcal{M}([n], \mathrm{rank}(\cdot))$ is defined by the set of integers $[n] = \{1, ..., n\}$ and the $\mathrm{rank}(\cdot)$ function, an integer valued funtion defined on all subsets of $[n]$ that satisfies the properties:

- $\mathrm{rank}(A) \geqslant 0$, for any $A \subseteq [n]$.
- $\mathrm{rank}(A) \leqslant |A|$, for any $A \subseteq [n]$.
- $\mathrm{rank}(A) \leqslant \mathrm{rank}(B)$, for any sets $A \subseteq B \subseteq [n]$.
- $\mathrm{rank}(A \cup B) + \mathrm{rank}(A \cap B) \leqslant \mathrm{rank}(A) + \mathrm{rank}(B)$, for any sets $A \subseteq B \subseteq [n]$.

A set A is called *independent* if $\mathrm{rank}(A) = |A|$; otherwise A is called dependent. A set is referred as a *circuit* if it is dependent *and all* of its proper subsets are independent. This means that if c is a circuit, then $\mathrm{rank}(c) = |c| - 1$.

Example: Let G be a $k \times n$ (e.g. a code generator) matrix over a field. Define the matroid $\mathcal{M}([n], \mathrm{rank}(\dot{)})$, where the rank of a set $A \subseteq [n]$ is $\mathrm{rank}(A) = \mathrm{rank}(G_A)$, G_A is the sub-matrix of G with columns indexed by A and rank operates on a set of columns in the well-known linear-algebraic way. In this case, the matroid $\mathcal{M}$ is *represented* by G.

B. Connections to Code Distance

A collection of sets $c_1, c_2, ...$ is said to have a non trivial union if every set is *not* contained in the union of the others, that is $c_i \nsubseteq \cup_{j \neq i} c_j$, for any i. Using the above definitions we state a simple lemma that will be fundamental in our derivations.

Lemma 1 *Let $c_1, ..., c_m$ be m circuits in $\mathcal{M}$. If the circuits have a non trivial union, then* $\mathrm{rank}(\cup_{i=1}^m c_i) \leqslant |\cup_{i=1}^m c_i| - m$.

Proof: We apply induction on m. For $m = 1$, since c_1 is a circuit $\mathrm{rank}(c_1) = |c_1| - 1$. Let $m > 1$ and denote by $c = \cup_{i=1}^{m-1} c_i$. By the property of the rank function $\mathrm{rank}(c \cup c_m) \leqslant \mathrm{rank}(c) + \mathrm{rank}(c_m) - \mathrm{rank}(c \cap c_m)$. Since the union of the circuits is non trivial $c \cap c_m$ is a proper subset of c_m and therefore is independent. Then by the induction assumption $\mathrm{rank}(c) + \mathrm{rank}(c_m) - \mathrm{rank}(c \cap c_m) \leqslant |c| - (m-1) + |c_m| - 1 - |c \cap c_m| = |c \cup c_m| - m$. ■

In what follows, we consider $\mathcal{M}$ to be the matroid that is represented by a code generator matrix G of size $k \times n$. We will define a new parameter μ relevant to the matroid $\mathcal{M}$, which will be used later in calculating the minimum distance of the code generated by G. We would like to note that μ can be defined also for non-representable matroids as well. We proceed with its definition and properties.

Definition 1 *Denote by μ the minimum integer such that the size of every non trivial union of μ circuits in $\mathcal{M}$ is at least $k + \mu$.*

Lemma 2 *Let μ be defined as above, then*

- *μ is bounded between 1 and $n+1$.*
- *There are μ circuits $c_1, ..., c_\mu$ in $\mathcal{M}$ whose union is nontrivial.*

Proof: Since there is *no* non trivial union of $n+1$ circuits, the statement: *any non trivial union of $n+1$ circuits is of size at least* $k + (n+1)$, is satisfied trivially, and therefore $\mu \leqslant n+1$. On the other hand, since a union of only one circuit is clearly a non trivial union, we conclude that $1 \leqslant \mu$. For the last part of the lemma, assume to the contrary that there are no μ circuits whose union is non trivial. Let μ' be the maximal integer such that there are μ' circuits $c_1, ..., c_{\mu'}$ whose union is non trivial. By the assumption $\mu' < \mu$, there exist μ' circuits $c_1, ... c_{\mu'}$ whose union is non trivial, and the size of the union is at most $k - 1 + \mu'$, namely

$$|\cup_{i=1}^{\mu'} c_i| \leqslant k - 1 + \mu'. \tag{4}$$

By the maximality of μ' we conclude that $\cup_{i=1}^{\mu'} c_i = [n]$, otherwise there would be a non trivial union of $\mu' + 1$ circuits. Hence, $k = \mathrm{rank}([n]) = \mathrm{rank}(\cup_{i=1}^{\mu'} c_i) \leqslant |\cup_{i=1}^{\mu'} c_i| - \mu' \leqslant k - 1 + \mu' - \mu' = k - 1$, where the first and the second inequalities follow from Lemma 1 and (4) respectively. This leads as to a contradiction, hence there are μ circuits whose union is non trivial. ■

The next theorem is the main result of this section. It characterizes the properties of the locality and minimum distance of a code in terms of the circuits in its matroid.

Theorem 2 *Let G, $\mathcal{M}$ and μ defined has above. Then,*

1) *the code has locality r iff each $i = 1, ..., n$ is contained in a circuit of size at most $r + 1$.*
2) *The distance of the code is equal to $d = n - k - \mu + 2$.*

Proof:

1) This follows trivially from the definition of a circuit.
2) If $\mu = 1$, then by definition, the size of any circuit is of size at least $k + 1$. Hence any k columns of G are linearly independent and G is a generator matrix of an MDS code, namely $d = n - k + 1$. If $\mu \geqslant 2$, then by the minimality of μ there exist $\mu - 1 \geqslant 1$ circuits $c_1, ..., c_{\mu-1}$ whose union is non trivial and is of size at most $k - 1 + \mu - 1 = k + \mu - 2$. Hence by Lemma 1 $\mathrm{rank}(\cup_{i=1}^{\mu-1} c_i) \leqslant |\cup_{i=1}^{\mu-1} c_i| - (\mu - 1) \leqslant k - 1$. Let x be a nonzero vector of length k which is orthogonal to the columns of G with indices in $\cup_{i=1}^{\mu-1} c_i$. Clearly such vector x exists since the rank of the union is at most $k - 1$. Then by the choice of x we get that $x \cdot G$ is a nonzero codeword of weight at most $n - (k + \mu - 2)$, and therefore also the minimum distance satisfies

$d \leqslant n - (k - \mu + 2)$. On the other hand, let T be the set of zero coordinates of some nonzero codeword from the code generated by G. Let $S \subseteq T$ be a maximal independent set in T. Clearly the size of S is at most $k-1$. Let $T \backslash S = \{t_1, ..., t_l\}$. We claim that $l \leqslant \mu - 1$. Assume the opposite, then for each $i = 1, ..., l$ the set $t_i \cup S$ contains a circuit that contains t_i, hence T contains at least l distinct circuits whose union is non trivial. From the definition of m we conclude that $k - 1 \geqslant |S| = |S \cup t_1 \cup ... \cup t_\mu| - \mu \geqslant k + \mu - \mu = k$, and we get a contradiction. The last inequality follows since $S \cup t_1 \cup ... \cup t_\mu$ contains μ circuits whose union is non trivial. Therefore the weight of the codeword is $n - |T| = n - (|S| + |T \backslash S|) \geqslant n - (k - 1 + \mu - 1)$. Hence also the minimum distance is at least $d \geqslant n - k - \mu + 2$, and the result follows. ■

From Theorem 2, we get the following corollary which characterizes all optimal linear LRCs.

Corollary 1 *The code generated by G has locality r, and optimal minimum distance $d = n - (k + \lceil \frac{k}{r} \rceil) + 2$ iff*

1) *Any $i = 1, ..., n$ is contained in a circuit of size at most $r + 1$.*
2) *The size of any nontrivial union of $\lceil \frac{k}{r} \rceil$ circuits in $\mathcal{M}$ is at least $k + \lceil \frac{k}{r} \rceil$.*

The previous corollary provided necessary and sufficient conditions for an optimal linear LRC. We derive from it another corollary which gives simple necessary conditions for optimal linear LRC. In what follows, we call a circuit nontrivial if its size is at most k.

Corollary 2 *Let G and $\mathcal{M}$ as before and let*

1) *all* nontrivial circuits *be of size at most $r + 1$ and let them form a partition of $[1, n]$.*
2) *For any collection of $\lceil \frac{k}{r} \rceil$ nontrivial circuits c_i $\left| \cup_{i=1}^{\lceil \frac{k}{r} \rceil} c_i \right| = \sum_{i=1}^{\lceil \frac{k}{r} \rceil} |c_i| \geqslant k + \lceil \frac{k}{r} \rceil$.*

Then the code has locality r, and optimal minimum distance $d = n - k - \lceil \frac{k}{r} \rceil + 2$.

IV. Optimality of Code Construction

In this section we will prove Theorem 1. For $i = 1, ..., m/r$, let V_i be the Vandermonde matrix of size $k \times r$ defined by the elements $\alpha_{1+r(i-1)}, ..., \alpha_{ir} \in \mathbb{F}_p$, namely $V_i = \begin{pmatrix} \overline{\alpha}_{r(i-1)+1} & \overline{\alpha}_{r(i-1)+2} & \cdots & \overline{\alpha}_{ir} \end{pmatrix}$. Then, we can rewrite the generator matrix G as

$$G = (V_1 \cdot A, V_2 \cdot A, ..., V_{m/r} \cdot A).$$

It is easy to check that any proper subset of the columns of A are linearly independent and therefore A also generates an $(r + 1, r)$ MDS code. This means that our code has locality r: any lost element can be repaired by accessing the r remaining elements that come from the same local code.

We continue by showing the optimality of its distance. By the construction, for any $i = 1, ..., n/(r + 1)$, the set of integers $c_i = [1 + (i - 1)(r + 1), i(r + 1)]$ forms a circuit of size $r + 1$ in the matroid represented by G. We will show that these circuits are the *only* circuits of size at most k, and by Corollary 2 this will imply the optimality of the minimum distance. Let S be all the k-subsets of $[n]$ that do *not* contain any circuit c_i, namely $S = \{s \subseteq [n] : |s| = k \text{ and } c_i \nsubseteq s \text{ for any } i = 1, ..., \frac{n}{r+1}\}$. For $s \in S$, denote by G_s the square sub-matrix of G restricted to columns with indices in s. Showing that $c_1, ..., c_{n/(r+1)}$ are the only circuits of size at most k is equivalent to showing that the matrix G_s is invertible for any $s \in S$. This will be done by showing that the determinant of G_s is a *nonzero* polynomial in a with coefficients in $\mathbb{F}_p$, and degree at most k. However, since a is a primitive element of the field $\mathbb{F}_{p^{k+1}}$, the degree of its minimal polynomial in $\mathbb{F}_p[x]$ is *exactly* $k + 1$. Therefore the determinant does not evaluate to zero in $\mathbb{F}_{p^{k+1}}$, *i.e.*, G_s is invertible for any $s \in S$, and the result will follow. To prove that, we first show a key property about the permanent of A. First, we extend the definition of a permanent of a square matrix to a non square matrix as follows. Let $B = (b_{i,j})$ be an $r \times t$ matrix and $t \leqslant r$, then

$$\text{perm}(B) = \sum_{(v_1, ..., v_t), v_i \neq v_j} \prod_{i=1}^{t} b_{v_i, i}. \tag{5}$$

Intuitively, the permanent is the sum of all products of elements in B, such that exactly one entry is picked from each column, and no two elements are picked from the same row. Note that throughout, the summation in (5) is done over $\mathbb{Z}$.

Lemma 3 *Let B be an $r \times t$ sub-matrix of A for $r \geqslant t$, then the permanent of B is a monic polynomial in a of degree at most t.*

Proof: Each of the products in (5) is a product of exactly t elements of B. In addition, each element equals to $a, 1$ or 0, hence the degree of each term is at most t. For the second part, note that B can be written as a block diagonal matrix with blocks $B_1, ..., B_m$, for some m. Hence the permanent of B is the product of the permanent of its blocks. Therefore the permanent of B is a monic polynomial, if the permanent of each block matrix is a monic polynomial. This fact can be easily verified, and the result follows. ■

For example, let B be composed of the columns $1, 2, 4$ and 5 of A of size 4×5 then

$$B = \begin{pmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 1 & a \end{pmatrix}.$$

Hence $B_1 = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$, and $B_2 = \begin{pmatrix} cca & 0 \\ 1 & a \end{pmatrix}$. Moreover $\text{perm}(B) = \text{perm}(B_1) \cdot \text{perm}(B_2) = 1 \cdot a^2 = a^2$.

Let G_s be a sub-matrix of G for some $s \in S$, and note that due to the structure of A each column in G_s is a linear combination of at most two columns of the form $\overline{\alpha}$. For example, let $r = 3, k = 6$. Moreover, if G_s is a 6×6 matrix composed of first three columns of $V_1 \cdot A$, first and third columns of $V_2 \cdot A$, and the second column of $V_3 \cdot A$. Then $G_s = (V_1 A_1, V_2 A_2, V_3 A_3)$, where for each $i = 1, 2, 3$ the matrix A_i is a sub-matrix of A. More precisely G_s can be written as

$$G_s = \begin{pmatrix} \overline{\alpha}_1^t & & & & & & & \\ a\overline{\alpha}_1^t & +\overline{\alpha}_2^t & & & & & & \\ & a\overline{\alpha}_2^t & +\overline{\alpha}_3^t & & & & & \\ & & & \overline{\alpha}_4^t & & & & \\ & & & & a\overline{\alpha}_5^t & +\overline{\alpha}_6^t & & \\ & & & & & & a\overline{\alpha}_7^t & +\overline{\alpha}_8^t \end{pmatrix}^t .$$

Recall that the determinant function is linear on its columns, namely if u and v are column vectors, B is some matrix, and α, β are scalars, then $\det(\alpha \cdot v + \beta \cdot u, B) = \alpha \cdot \det(v, B) + \beta \cdot \det(u, B)$. Therefore the determinant of G_s can be expanded into a linear combination of powers of a multiplied by determinants of Vandermonde matrices, *i.e.,* it is a polynomial in a over $\mathbb{F}_p$. The degree of the polynomial $\det(G_s)$ is two, and its leading coefficient equals to $\det(\overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_3, \overline{\alpha}_4, \overline{\alpha}_5, \overline{\alpha}_7) \in \mathbb{F}_p$. This coefficient is non zero because all the α_i's are distinct. Note that there are other terms in the expansion that could potentially contribute a greater degree of a, such as

$$a^4 \cdot \det(\overline{\alpha}_1, \overline{\alpha}_1, \overline{\alpha}_2, \overline{\alpha}_4, \overline{\alpha}_5, \overline{\alpha}_7). \tag{6}$$

However, this term equals to zero since $\overline{\alpha}_1$ appears twice in (6). It is easy to see that in the expansion of the polynomial $\det(G_s)$ there is exactly *one non zero* term of the form $c \cdot a^2$ for some $c \in \mathbb{F}_p$, hence, $\det(G_s)$ is a non zero polynomial in a of degree 2.

In the general case, G_s can be written as $G_s = (V_{i_1}A_1, ..., V_{i_t}A_t) = (V_{i_1}, ..., V_{i_t}) \cdot D(A_1, ..., A_t)$ for some $1 \leqslant t \leqslant m/r$, and $D = D(A_1, ..., A_t)$ is a block diagonal matrix with the matrices A_i along its diagonal. Furthermore, each A_i is a sub-matrix of A. Clearly, $\text{perm}(D)$ equals the product of the permanent of its blocks and its degree equals to the degree of $\det(G_s)$. Moreover, the coefficient of a^i in $\text{perm}(D)$ indicates the number of nonzero terms of degree i in the expansion of $\det(G_s)$. By Lemma 3, each of the polynomials $\text{perm}(A_i)$ is monic, hence also $\text{perm}(D)$: there is only one non zero term in the expansion of $\det(G_s)$ with the largest degree of a. Therefore $\det(G_s)$ is a nonzero polynomial in a over $\mathbb{F}_p$. Moreover, D has k columns, therefore the determinant is a non zero polynomial of degree at most k. For the final step, since the minimal degree of a non zero polynomial over $\mathbb{F}_p$ that annihilates a is $k+1$, we conclude that $\det(G_s)$ is a non zero number in $\mathbb{F}_{p^{k+1}}$, and therefore G_s is invertible.

V. Conclusions

In this work we introduced a new family of optimal (n, k, r)-LRCs that are simple to implement. The codes are based on re-encoding Reed-Solomon encoded blocks for the added property of locality. To prove the optimality of our code, we first establish a connection between the minimum code-distance and properties on a matroid represented by the generator matrix of a code. We continue by showing that some subsets of the columns in the code generator matrix are full-rank. Our code construction is simple and requires a large finite field. This, however, does not seem to be a significant practical problem since each field element requires $O(k \log n)$ bits to be represented. Explicit constructions of optimal LRCs for the case when $r+1$ does not divide n and for small finite fields remain as open problems.

VI. Acknowledgement

This research was supported in part by NSF grant CCF1217894. Moreover we would like to thank Uzi Tomo Magen for intriguing and useful discussions.

References

[1] H. Weatherspoon and J. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," *Peer-to-Peer Systems*, pp. 328–337, 2002.

[2] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in windows azure storage," in *USENIX Annual Technical Conference (USENIX ATC)*, 2012.

[3] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: Novel erasure codes for big data," *Proceedings of the VLDB Endowment (to appear)*, 2013.

[4] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *Information Theory, IEEE Transactions on*, vol. 56, no. 9, pp. 4539–4551, 2010.

[5] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the msr and mbr points via a product-matrix construction," *Information Theory, IEEE Transactions on*, vol. 57, no. 8, pp. 5227–5239, 2011.

[6] C. Suh and K. Ramchandran, "Exact-repair mds code construction using interference alignment," *Information Theory, IEEE Transactions on*, vol. 57, no. 3, pp. 1425–1442, 2011.

[7] I. Tamo, Z. Wang, and J. Bruck, "Mds array codes with optimal rebuilding," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pp. 1240–1244, IEEE, 2011.

[8] V. R. Cadambe, C. Huang, S. A. Jafar, and J. Li, "Optimal repair of mds codes in distributed storage via subspace interference alignment," *arXiv preprint arXiv:1106.1250*, 2011.

[9] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through hadamard designs," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pp. 1382–1389, IEEE, 2011.

[10] O. Khan, R. Burns, J. Plank, and C. Huang, "In search of i/o-optimal recovery from disk failures," in *Proceedings of the 3rd USENIX conference on Hot topics in storage and file systems*, pp. 6–6, USENIX Association, 2011.

[11] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *Information Theory, IEEE Transactions on*, vol. 58, no. 11, pp. 6925–6934, 2011.

[12] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *INFOCOM, 2011 Proceedings IEEE*, pp. 1215–1223, IEEE, 2011.

[13] D. S. Papailiopoulos, J. Luo, A. G. Dimakis, C. Huang, and J. Li, "Simple regenerating codes: Network coding for cloud storage," in *INFOCOM, 2012 Proceedings IEEE*, pp. 2801–2805, IEEE, 2012.

[14] J. Han and L. A. Lastras-Montano, "Reliable memories with subline accesses," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pp. 2531–2535, IEEE, 2007.

[15] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," in *Network Computing and Applications, 2007. NCA 2007. Sixth IEEE International Symposium on*, pp. 79–86, IEEE, 2007.

[16] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pp. 2771–2775, IEEE, 2012.

[17] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar, "Optimal linear codes with a local-error-correction property," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pp. 2776–2780, IEEE, 2012.

[18] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration," *arXiv preprint arXiv:1211.1932*, 2012.

[19] A. Rawat, O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *arXiv preprint arXiv:1210.6954*, 2012.

[20] A. S. Rawat and S. Vishwanath, "On locality in distributed storage systems," *arXiv preprint arXiv:1204.6098*, 2012.